Mentat IPv6 Implementation
Status and Experience

December 7, 1995

IP[v6] Stack Background

TCP/IPv4 STREAMS product, stack & basic APIs

Additional STREAMS modules and minor modifications to existing modules

Customers “drop in” new v4/v6 stack

No impacts on existing IPv4 applications

Implementation Status

v6 via v4 tunneling interoperation:

- TCP Expo @ San Jose (8/95) - DEC, HP/SICS, INRIA
- Interop @ Atlanta (9/95) - DEC, HP/SICS, INRIA

TCP[v6] applications demonstrated, UDP[v6] with in-house tests

Latest Neighbor Discovery (ND) tested in-house

DAD & NUD underway from autoconfig & ND specs

Packet forwarding using manually configured addresses

Platforms Supported by Prototype

Displayed publicly, running v4 & v6 simultaneously:

- Windows/NT - ftp client/server, telnet client, Web client
- MacOS - ftp & Web client/server, telnet client, ping
- Motorola SVR4 - ftp & telnet client/server, ping

In-house - 2 other UNIX platforms with subset of above

1 OEM customer experimenting, another committed

Specifications Implementation

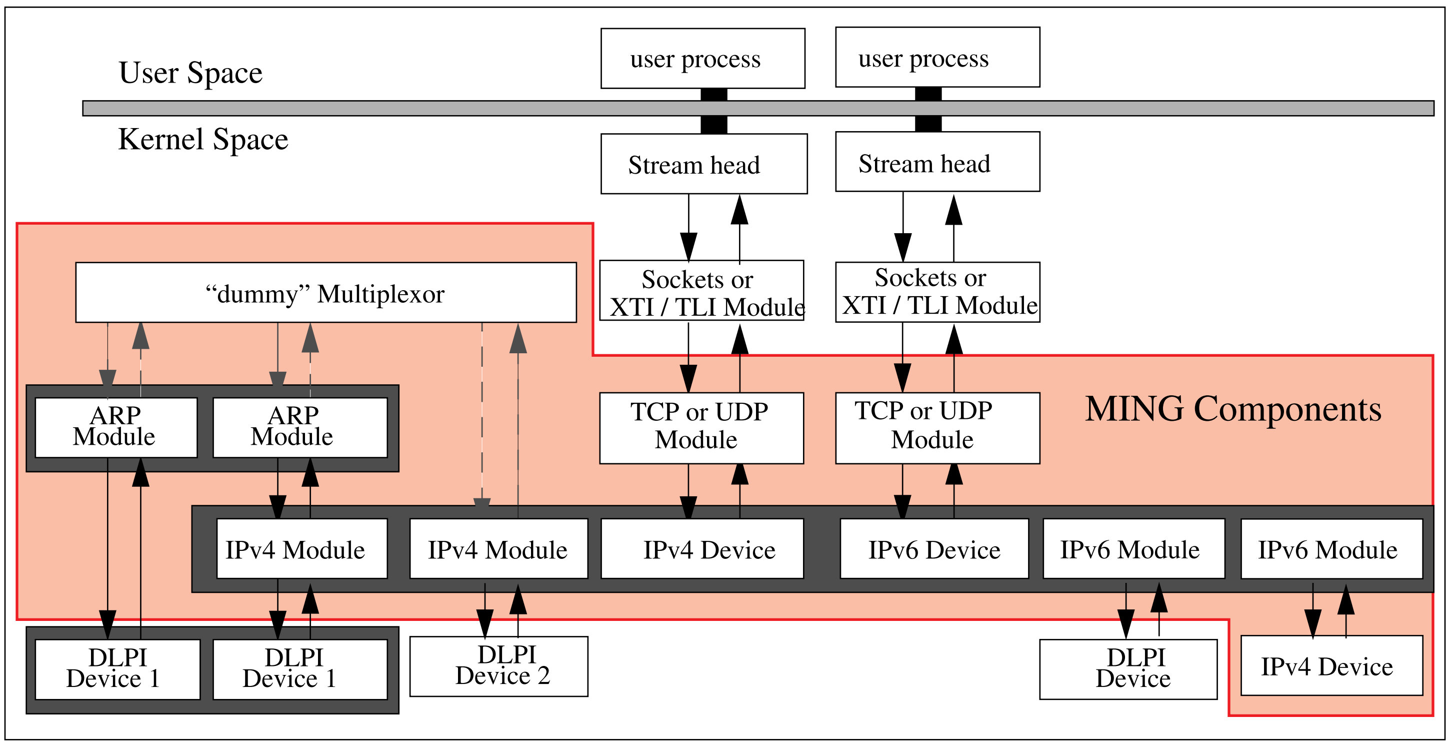
At least partially

- addressing specs
- autoconfig
- bsd-api
- discovery
- ethernet-networks
- fddi-networks
- icmp
- ipv6-spec
- ngtrans-mech

To be done (at least kernel portions)

- Security
- DHCPv6
- DNS

IPv4/IPv6 Streams Architecture



Implementation Points

Parts of IPv4 stack made common, IPv6 hooks added

- Inbound/outbound IP version # checks
- ULP dual-address support

Ethernet type provided architecture flexibility

- Possible v6 into non-Mentat v4 stack platform

v6 STREAMS architecture/plumbing simpler

- ND, unlike v4 ARP, doesn't need unique DLPI bind

More burden on drivers to (efficiently) support multicast

Implementation Points (cont'd)

ULP ports global across AF_INET[6]

- Simple to convert monolithic servers
- Lose some v6-specific application deployment options (demultiplexing inbound packets)

TCP, UDP interface to application work significant but manageable

- v4 & mapped addresses to select v4 vs. v6 stacks
- Socket options differences - *some* map directly
- Source routes specified differently

Carry full 16-byte addresses internally, even for v4 apps

Implementation Points (cont'd)

Translation module for binary compatibility

- STREAMS flexibility permits API "conversion"
- v4 application clients, as well as servers, speak v4 or v6
- Several unmodified binary v4 applications to operate via v6
- Aided v6 interops, also compensated for varying interpretations of FTP over v6
  - → (This was/is the LPORT vs. PORT issue.)
- Currently supports system-wide manual configuration
  - → Current specifiers are ports & addresses
- Scalability and management issues
  - → Futures - DNS hooks, bogus v4 tokens, security

General comments

Main-line specs are looking good for stack implementation

Lots of items, such as tunneling, just "make sense"

Recently available v6 routers should aid further testing

Kernel interface "de facto" APIs would be nice

- Aid portability of experimental routing daemons, utilities
- Currently assuming BSD 4.3-like as much as possible